DOLBY ATMOS – LIVE WORKFLOW

AWS Elemental Live – AWS Elemental MediaStore – Amazon CloudFront

CONTENT

INTRODUCTION

This workflow example shows how to use AWS Elemental Live, AWS Elemental MediaStore and Amazon CloudFront services to deliver HLS and DASH streams with Dolby Atmos audio.

AWS Elemental Live is an on-premises solution that processes live video for delivery to broadcast televisions and streaming to internet-connected devices. It allows you to easily and reliably compress live video sources into multiple versions for distribution. Deployed on-premises as appliances or as AWS-licensed software on your infrastructure, AWS Elemental Live lets you deploy reliable, high-quality live video processing at the source, within the studio, or inside remote facilities. For more information see https://aws.amazon.com/elemental-live/

AWS Elemental MediaStore is a video origination and storage service. It offers the performance and consistency required for delivering streaming media combined with the security and durability AWS offers across its services. The service can be accessed through the AWS Management Console, or via APIs or AWS SDKs and configured to your workflow. For more information see https://aws.amazon.com/mediastore/

Amazon CloudFront is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency, high transfer speeds, all within a developer-friendly environment. CloudFront is integrated with AWS – both physical locations that are directly connected to the AWS global infrastructure, as well as other AWS services. For more information see https://aws.amazon.com/cloudfront/

REQUIREMENTS

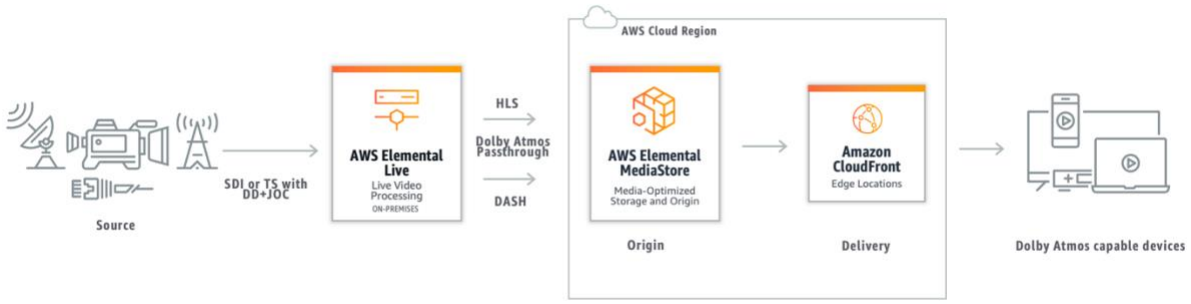
To follow this workflow, you need:

- An AWS Elemental Live encoder (version 2.16.3 or higher) connected to the internet
- An SDI or TS source with Dolby Atmos audio Metadata
- An AWS account for AWS Elemental MediaStore and Amazon CloudFront
- Credentials/stream keys for delivery to your MediaStore container
- A Dolby developer account (this is free of charge)

ORDER OF WORK

Step A: Create a container in AWS Elemental MediaStore with Amazon CloudFront
Step B: Create an event in AWS Elemental Live
Step C: Start streaming the video: Dolby Atmos in AWS Elemental Live
Step D: Validate your HLS or DASH Dolby Atmos stream

Workflow: AWS Elemental Live – AWS Elemental MediaStore -Amazon CloudFront



STEP A: Create a container in AWS Elemental MediaStore allowing Amazon CloudFront distribution

1. Open the MediaStore console at https://console.aws.amazon.com/mediastore/
2. On the **Containers** page, choose the container name.

   The container details page appears.

3. In the **Container policy** section, attach a policy that grants read access or greater to Amazon CloudFront. *Example container policy: Public read access over HTTPS*

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadOverHttps",
      "Effect": "Allow",
      "Action": ["mediastore:GetObject", "mediastore:DescribeObject"],
      "Principal": "*",
      "Resource": "arn:aws:mediastore:<region>:<owner acct number>:container/<container name>/*",
      "Condition": {
        "Bool": {
          "aws:SecureTransport": "true"
        }
      }
    }
  ]
}
```

   Note: Make sure that you add the right *<region>:<owner acct number> <container name>*

4. In the **Container CORS policy** section, assign a policy that allows the appropriate access level. A CORS policy is necessary in this case since you want to provide access to a browser-based player.

```
[
  {
    "AllowedOrigins": [
      "*"
    ],
    "AllowedMethods": [
      "GET",
      "HEAD"
    ],
    "AllowedHeaders": [
      "*"
    ],
    "MaxAgeSeconds": 3000
  }
]
```

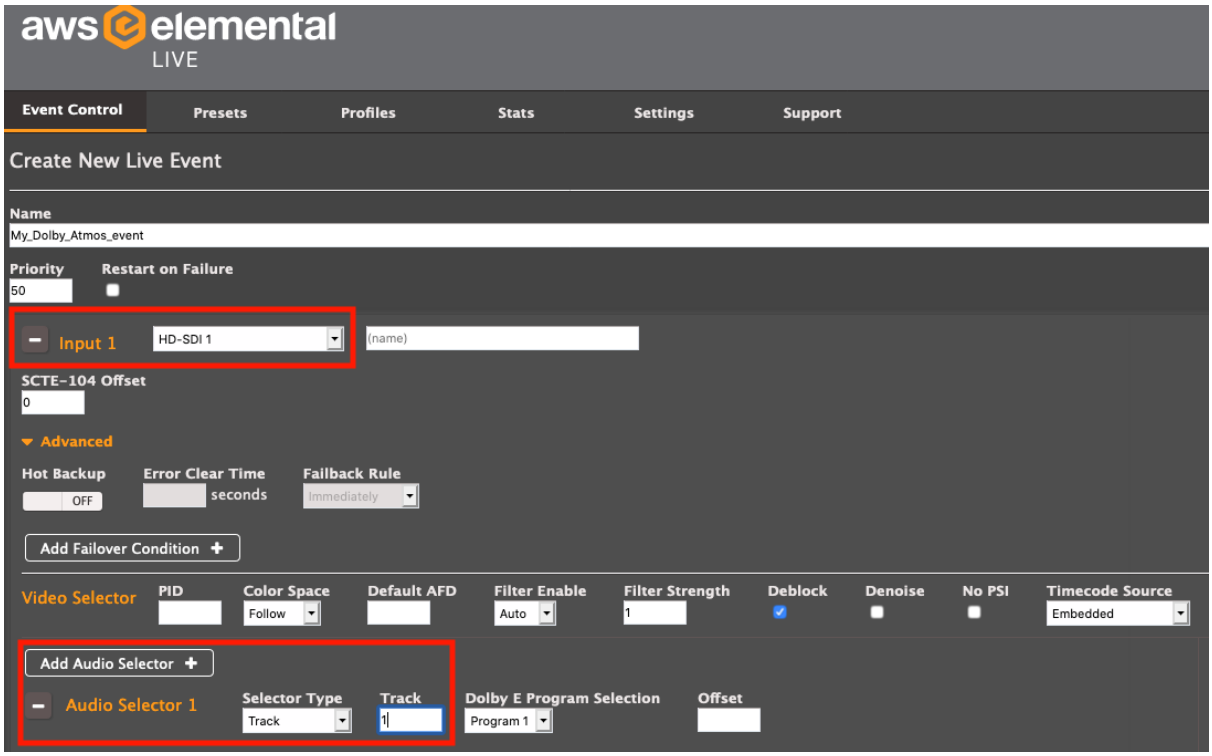
5. Make note of the following details:

- The data endpoint that is assigned to your container. You can find this information in the **Info** section of the **Containers** page. In CloudFront, the data endpoint is referred to as the *origin domain name*.
- The folder structure in the container where the objects are stored. In CloudFront, this is referred to as the *origin path*. Note that this setting is optional. For more information about origin paths, see the Amazon CloudFront Developer Guide.

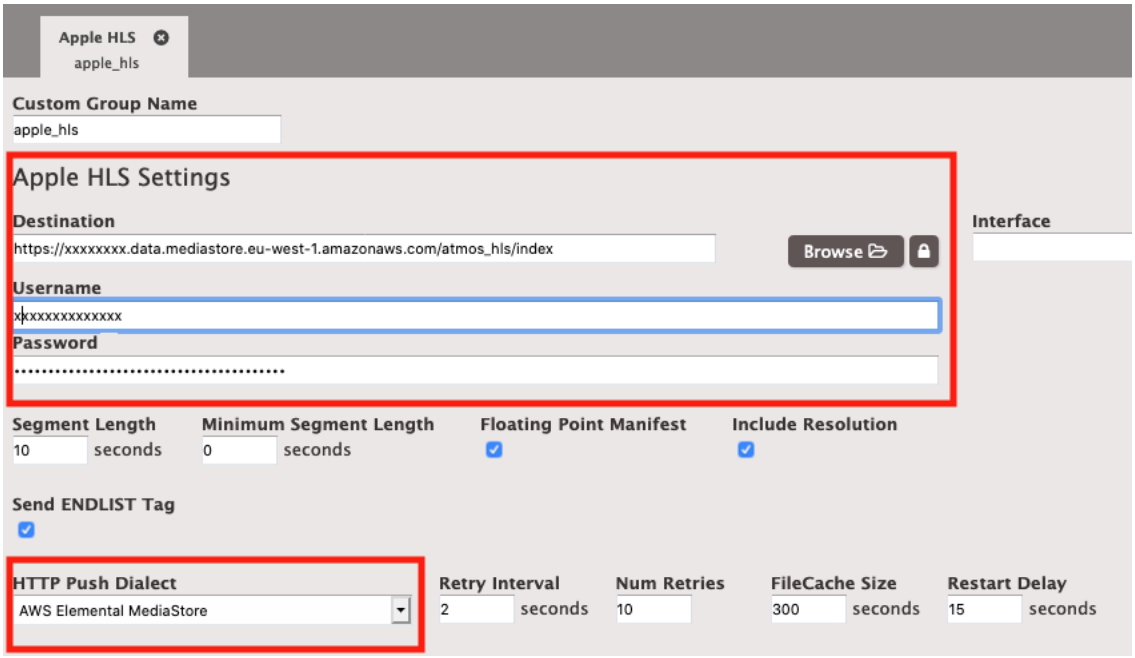
  In CloudFront, create a distribution that is configured to serve content from AWS Elemental MediaStore. You will need the information that you collected in the preceding step. Wait until the “Status” changes to “Deployed”.

STEP B: Create an event in AWS Elemental Live

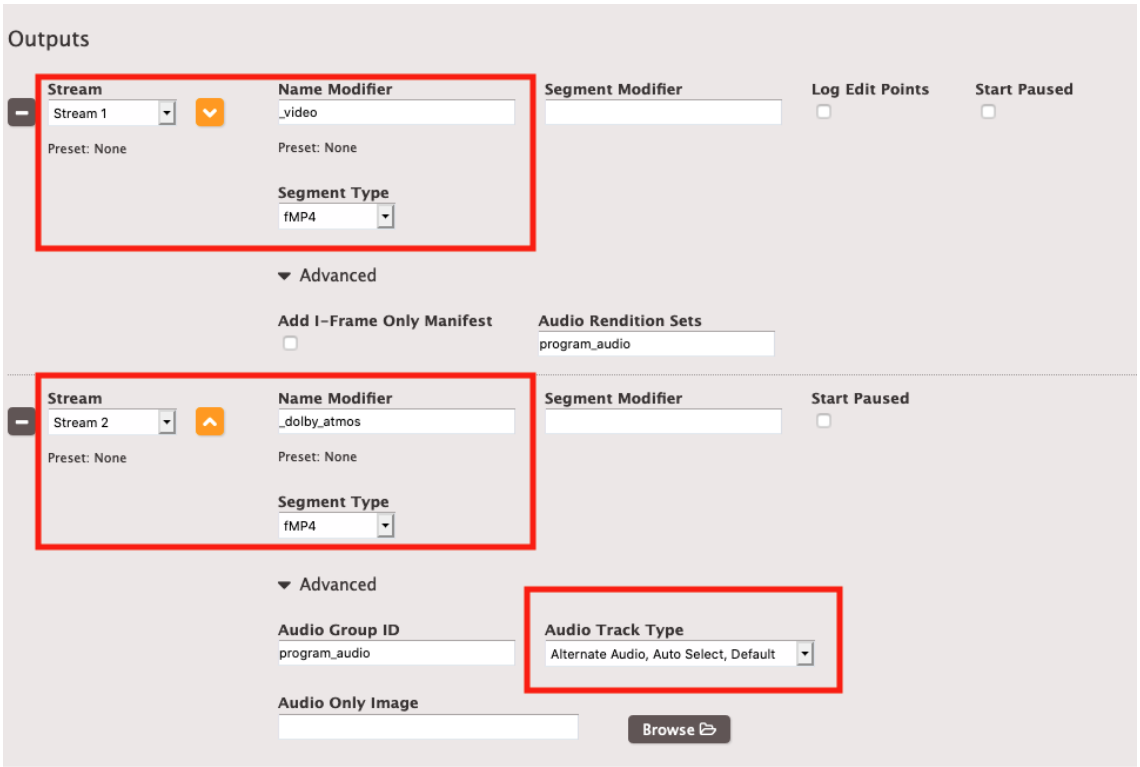
1. In Elemental Live select **new event**.
2. In the **input side**, specify an event **name** and select the type of **input**. Dolby Atmos can be passthrough from an SDI or TS input. For this example, **SDI** has been selected.
3. Select the **Audio Selector 1** and specify the **selector type** and **track** where your Dolby Atmos is mapped. For this example, Dolby Atmos is mapped in **track 1**.



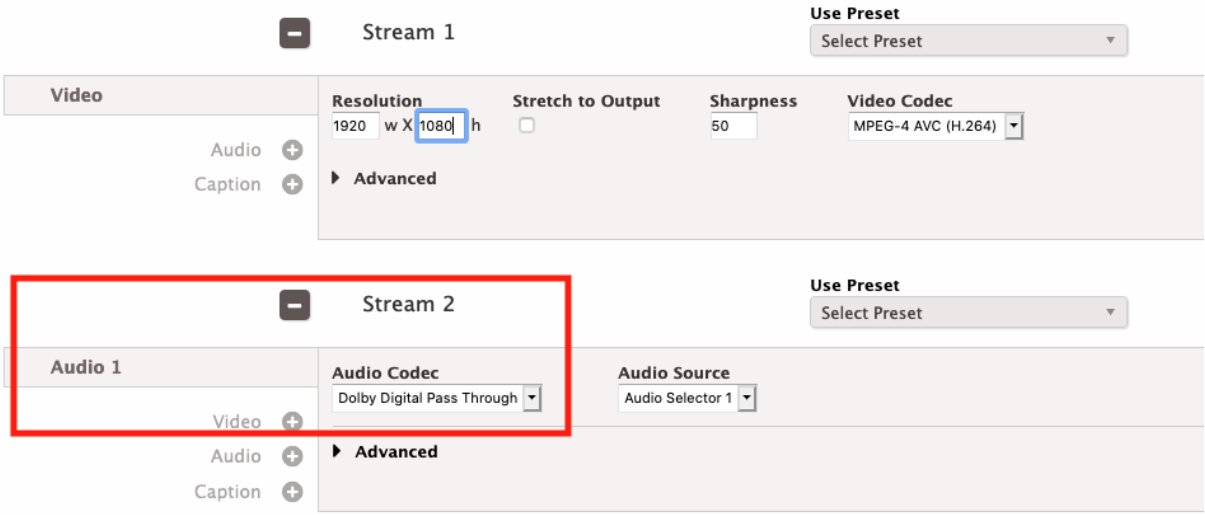
4. In the output side, select the **output group** that you would like to use. For this example, **HLS CMAF** has been selected.
5. Introduce your **Mediastore URL** in the **destination field** and also add your **AWS Access Key and Secret Access Key** associated to your MediaStore container into the **"Username and Password"** fields.
6. Select **AWS Elemental MediaStore** as **HTTP Push Dialect**.



7. Add an output for the **Video** and output for the **Audio**. HLS CMAF requires that the Video and Audio streams are separated. Otherwise, a validation error will pop-up when you try to start the event. Give a name to each output and select **"fMP4" as segment type**



8. Configure your **video elementary stream** as you wish.
9. Configure your **audio elementary stream as Dolby Digital Passthrough** and select the **Audio Source**.

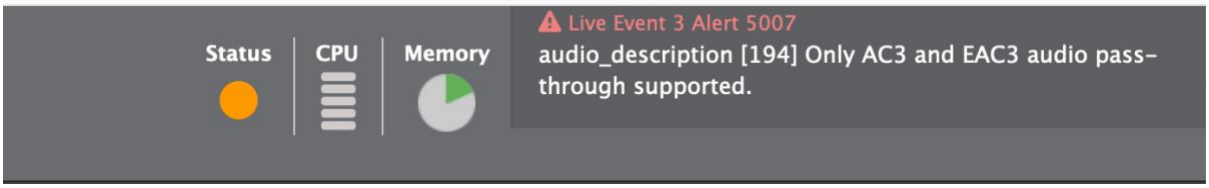


10. Now the event configuration is completed, you can **start the event**.

STEP C: Start the event: Dolby Atmos in AWS Elemental Live

Currently, AWS Elemental Live supports Dolby Atmos (DD+JOC) Passthrough. You can pass through Dolby Atmos from an SDI or TS input to MP4 in Archive, DASH, HLS, and UDP/TS output groups. The steps below help to verify that your source contains Dolby Atmos audio:

1. If your audio input selection does not contain Dolby Metadata or you selected the wrong track/PID you will receive an alert to let you know.



2. In the Elemental encoder once the channel has started, go into the **Elemental event logs** and select the **eme.log**.
3. There you can check for the **EAC3+JOC metadata**. Search for "**Audio Decoder "initialized**.
4. Characteristic of the Dolby Atmos metadata:
    - Dolby Atmos audio is carried by EAC3 codec.
    - Dolby Atmos is usually encoded in **bitrates between 384 and 768 kbps.**
    - Dolby Atmos complexity_index_type_a field takes a value of **1 to 16** that indicates the decoding complexity of the Dolby Atmos bitstream.

Please see below an example from the Elemental logs:

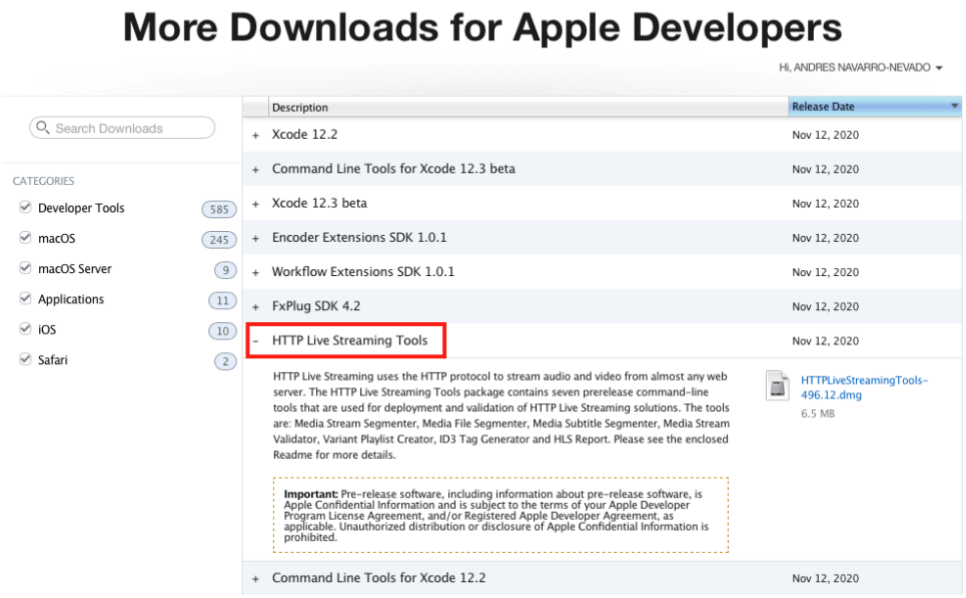
I [71215] {3,global} [AudioDecoder selector-3-track-1-pt-drc] thread started. Thread id is [71215]; priority [tpNORMAL]; scheduling [SCHED_RR]. Bound to [16] CPUs FirstCPU [0]
I [71215] {3,global} Audio decoder [selector-3-track-1-pt-drc] configured to use audio codec [PassThrough]
I [71215] {3,global} PASSTHROUGH audio decoder initialized: [ec3] [10]channel(s), [N/A]coding mode, [448]kbps, [48000]samples/sec, [10]bytes/frame
I [71168] {3,global} Passing through audio stream [selector-3-track-1-pt-drc] from: [EAC3], [16] channels, [S16] sample format, [48000] samples/sec, [448000] bps.
I [71168] {3,global} Audio pipeline started, input [3]

STEP D: Validate your HLS or DASH Dolby Atmos stream

Apple HTTP Live Streaming Tools (MAC users):

1. Create an **account** in Apple Developer.com, it's a free site (simply register): https://developer.apple.com

2. Sign in with the **Apple ID** associated with your developer account. On the left-hand side, click in **Download**. A new page will open, there select **"More"** and download the **HTTP Live Streaming Tools.**



3. Once downloaded, install the **HTTP Live Streaming Tools** software in your MAC. Then open a **terminal** and type man mediastreamvalidator. The MediaStreamValidator manual will be displayed.

```
mediastreamvalidator(1)   BSD General Commands Manual  mediastreamvalidator(1)

NAME
     mediastreamvalidator -- Validates HTTP Live Streaming streams and servers.

SYNOPSIS
     mediastreamvalidator [-T | --description=string] [-d | --device=(ipod | iphone | atv | ipad)] [-h | --help] [-i | --immediate] [-p | --parse-playlist-only]
                          [-t | --timeout=duration] [-u | --userAgent=string] [-O | --validation-data-path=path] [-V | --verbose] [-v | --version] url

DESCRIPTION
     The mediastreamvalidator is a command-line tool that validates HTTP Live Streaming streams and servers. It generally operates on a playlist URI or local
     path.

OPTIONS
     -T | --description=string
           Adds the description string to the JSON and console output.  Usually a description of the stream being validated.

     -d | --device=(ipod | iphone | atv | ipad)
           Specify the device whose user agent will be used in HTTP communication. Possible values are ipod, iphone, atv or ipad.

     -h | --help
           Show help

     -i | --immediate
           Output errors immediately after they are detected. Most useful with live streams.

     -p | --parse-playlist-only
           Only parses the playlist.

     -t | --timeout=duration
           Specifies validation timeout in seconds. Default timeout is 300 seconds (5 minutes).

     -u | --userAgent=string
           Sets the user agent string to be used in HTTP communication. Default user agent is set at runtime if one was not set.

     -O | --validation-data-path=path
           Logs validation data to JSON file at the specified path.

     -V | --verbose
           Enable verbose output. Default is off.

     -v | --version
           Show version number.

EXAMPLES
     mediastreamvalidator -t 10 http://example.com/playlist.m3u8

     Validates a remote playlist. Validation halts after 10 second timeout is reached.

     mediastreamvalidator -p http://example.com/playlist.m3u8

     Only parses a remote playlist.

     mediastreamvalidator -d atv http://example.com/playlist.m3u8

     Validates a remote playlist. Uses a user agent equivalent to that used by the Apple TV during HTTP Communication.

SEE ALSO
     hlsreport(1), mediastreamsegmenter(1)
```

4. Validate the HLS Dolby Atmos stream.
   Select the Cloudfront URL + manifest name. In this sample the stream will be validated against Apple TV:
   $ mediastreamvalidator -d atv https://xxxxxxxxx.cloudfront.net/index.m3u8

   Result:

---
index_dolby_atmos.m3u8

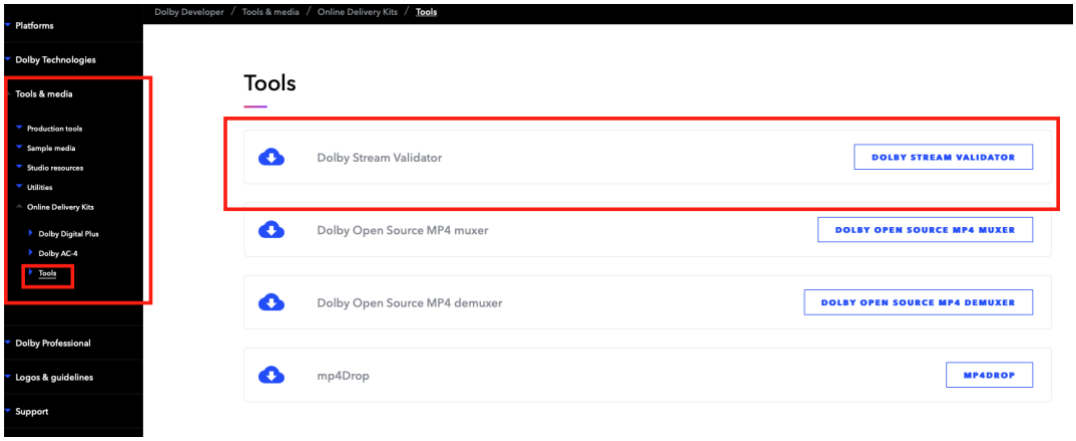
HTTP Content-Type: application/vnd.apple.mpegurl

Processed 6 out of 6 segments
Average segment duration: 9.920000
Total segment bitrates (all discontinuities): average: 643.14 kb/s, max: 643.26 kb/s
Rendition group ID: program_audio_0

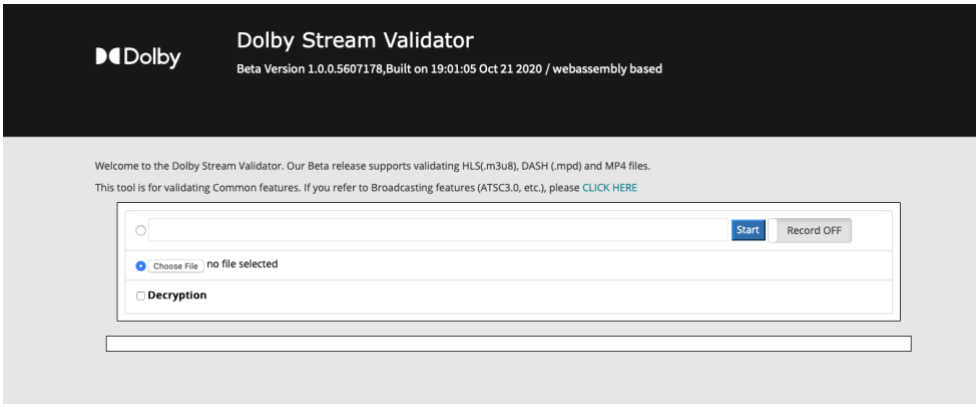
Discontinuity: sequence: 0, parsed segment count: 6 of 6, duration: 59.520 sec, average: 643.14 kb/s, max: 643.26 kb/s
Track ID: 1
Audio Codec: ec+3
Audio sample rate: 48000 Hz
Audio channels: 16
Audio channel layout: (null)

Dolby Developer site:

1. Create an **account** in Dolby Developer.com, it's a free site (simply register): https://developer.dolby.com/
2. On the left-hand side, navigate to **"Tools & Media"** and select **"Online Delivery Kits"**. Click in **"Tools"**, a new page will open.



3. Access the **Dolby Stream Validator**. This tool supports validating HLS(.m3u8), DASH (.mpd) and MP4 files.



4. Check Dolby Atmos stream.

   Add your **CloudFront URL + manifest name**. In this sample:

   HLS: https://xxxxxxxxx.cloudfront.net/index.m3u8

   DASH: https://xxxxxxxxx.cloudfront.net/index.mpd

   Below is an example of how Dolby Atmos is signalled on HLS and DASH content:

HLS

Audio Track: Codec: EC3, Channels: 6, Language: und, JOC. Validation result: PASSED

<HLS>
<EXT-X-VERSION>4</EXT-X-VERSION>
<EXT-X-STREAM-INF BANDWIDTH="5837332" AVERAGE-BANDWIDTH="6204000" CODECS="avc1.4d401f,ec-3" RESOLUTION="1280x720" FRAME-RATE="25.000" AUDIO="program_audio_0">
https://xxxxxxxx.cloudfront.net/index_video.m3u8
</EXT-X-STREAM-INF>
<EXT-X-MEDIA TYPE="AUDIO" LANGUAGE="eng" NAME="Alternate Audio" AUTOSELECT="YES" DEFAULT="YES" CHANNELS="16/JOC" GROUP-ID="program_audio_0"URI="https://xxxxxxxx.cloudfront.net/index_dolby_atmos.m3u8"/>
</HLS>

DASH

Audio Track: Codec: EC3, Channels: 6, Language: und, JOC. Validation result: PASSED

<MPD xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="urn:mpeg:dash:schema:mpd:2011" xmlns:cenc="urn:mpeg:cenc:2013" xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 http://standards.iso.org/ittf/PubliclyAvailableStandards/MPEG-DASH_schema_files/DASH-MPD.xsd" type="static" mediaPresentationDuration="PT32.400S" availabilityStartTime="2020-10-27T20:57:59.388Z"minBufferTime="PT32.400S" suggestedPresentationDelay="PT1M4.800S" profiles="urn:mpeg:dash:profile:isoff-live:2011">
<Period start="PT0S" id="1">
<AdaptationSet mimeType="video/mp4" frameRate="25/1" segmentAlignment="true" subsegmentAlignment="true" startWithSAP="1" subsegmentStartsWithSAP="1" bitstreamSwitching="false">
<SegmentTemplate timescale="90000" media="index_dash_video_$Number%09d$.mp4" initialization="index_dash_videoinit.mp4" duration="2916000" startNumber="1"/>
<Representation id="1" width="1280" height="720" bandwidth="5000000" codecs="avc1.4d401f"/>
</AdaptationSet>
<AdaptationSet mimeType="audio/mp4" segmentAlignment="0">
<SegmentTemplate timescale="48000" media="index_dash_audio_$Number%09d$.mp4" initialization="index_dash_audioinit.mp4" duration="1555200" startNumber="1"/>
<Representation id="2" bandwidth="640000" audioSamplingRate="48000" codecs="ec-3">
<AudioChannelConfiguration schemeIdUri="urn:mpeg:mpegB:cicp:ChannelConfiguration" value="6"/>
<SupplementalProperty schemeIdUri="tag:dolby.com,2018:dash:EC3_ExtensionType:2018" value="JOC"/>
<SupplementalProperty schemeIdUri="tag:dolby.com,2018:dash:EC3_ExtensionComplexityIndex:2018" value="16"/>
</Representation>
</AdaptationSet>
</Period>
</MPD>

STEP E: CLEAN UP

Once your test is finished, to avoid additional charges, it's important to stop and delete all of the AWS resources you used.

1. In the AWS CloudFront console, under CloudFront Distributions, choose the **distribution ID** that you wish to end, then choose **disable** (this process can take up to 15 mins) and wait for the status to change from "In Progress" to "Disable".
2. Select again the **distribution ID** and choose **Delete.**
3. Once the CloudFront distribution has been deleted access to the **AWS MediaStore console**, select the **container** that you wish to delete and click **Delete**. Note: That this action permanently deletes the container.